

ТЕХНОЛОГИИ ПРИМЕНЕНИЯ JDOM-ФУНКЦИЙ ДЛЯ ОРГАНИЗАЦИИ ВЗАИМОДЕЙСТВИЯ JAVA И ПРИЛОЖЕНИЙ MICROSOFT НА ПРИМЕРЕ EXCEL

Фролов А.В., Трофимов С.П.

Уральский федеральный университет имени первого Президента России Б.Н.Ельцина, Екатеринбург, ИРИТ-РТФ

В настоящее время актуальными являются проблемы взаимодействия программ, написанных на разных языках программирования. Как, например, из Visual Basic вызвать функции некой DLL, написанной на C. Также необходимо обеспечивать взаимодействие между различными приложениями. В данном докладе рассматривалось взаимодействие Java-приложений и табличного процессора Excel, особенно популярном в образовательном пространстве. Данная проблема решалась в ходе написания курсового проекта «Использование JDOM-функций для работы с научными данными в XML-документах на языке Java». В результате работы была написана программа на языке Java, взаимодействующая посредством XML-файлов с табличным процессором Excel. Программа решает задачу оптимизации и записывает результаты её решения в другой XML-файл. Для решения задачи оптимизации в Excel использовался сервис «Поиск решения», вызов которого из программы на Java реализован в виде vbs-скрипта.

Ключевые слова: программа, взаимодействие, приложение, макрос, «Поиск решения», XML (расширяемый язык разметки), Excel, DOM (объектная модель документа), Java, VBS (Visual Basic Script), JDOM (Java DOM).

TECHNOLOGIES OF JDOM-FUNCTIONS USING FOR ORGANISATION OF INTERACTION JAVA AND MICROSOFT APPLICATIONS FOR EXAMPLE EXCEL

Nowadays the problems of programs interaction are actual, which are written on different programming languages. For instance, some DLL function written in C is called from Visual Basic. Also it is necessary to provide interaction between different applications. In this report interaction between Java-applications and Excel spreadsheet is considered, especially popular in education field. This problem was solving within writing of course work “JDOM-functions usage for work with scientific data in XML-files on Java programming language”. As a result of the work the program was written in Java, interacting by means of XML-files with Excel spreadsheet, which solves optimization task, and writing results of its solution to the another XML-file. To solve the optimization task in Excel the service Solver was used, whose call from Java program was implemented as the vbs-script.

Key words: program, interaction, application, macros, “Solver”, XML (eXtensible Markup Language), Excel, DOM (Document Object Model), Java, VBS (Visual Basic Script), JDOM (Java DOM).

Целью работы являлось получение удобного механизма обработки и хранения данных, которые структурированы и записаны в формате XML-файла. В качестве предметной области взята экономическая задача, связанная с оптимизацией использования ресурсов для получения максимальной прибыли. После формализации предметной области, нами составлена программа, которая помещает в XML-файл входные и выходные данные, а также служебные данные, поясняют принцип взаимодействия разрабатываемого приложения с Excel-файлом, в котором будет происходить решение данной задачи. Последний способ использования XML-файла, предназначенный для передачи данных между приложениями, получил широкое применение в наши дни.

XML обеспечивает стандартный способ разметки контента (содержания), предоставляя гибкий способ структурирования данных. Контент XML документа

размечается с помощью определяемых разработчиком тегов, на основе правил грамматики XML.

Microsoft Excel предоставляет широкий спектр использования электронных таблиц для различных областей науки и бизнеса. Электронные таблицы в свою очередь представляют собой удобный инструмент для автоматизации вычислений. Многие расчёты, в частности в области бухгалтерского учёта, выполняются в табличной форме: балансы, расчётные ведомости, сметы расходов и т. п. Кроме того, решение численными методами целого ряда математических задач удобно выполнять именно в табличной форме. Использование математических формул в электронных таблицах позволяет представить взаимосвязь между различными параметрами некоторой реальной системы. Решения многих вычислительных задач, которые раньше можно было осуществить только с помощью программирования, стало возможно реализовать через математическое моделирование в электронной таблице.

Вышеописанные возможности стали доступны благодаря наличию в Excel таких сервисов, как «Пакет анализа» и «Поиск решения», функционал последнего используется в этой работе. Например, «Пакет анализа» содержит 19 статистических процедур и около 50 функций, которые в основном относятся к категориям инженерных и финансовых. Статистические процедуры представляют широкий спектр статистического анализа, начиная от простой описательной статистики или сглаживания данных и заканчивая анализом Фурье и проведением различных тестов. «Поиск решения» позволяет определить, при каких значениях указанных влияющих ячеек формула в целевой ячейке принимает нужное значение (минимальное, максимальное или равное какой-либо величине). Для расчета заданного значения применяются различные математические методы поиска. В качестве примера совместного применения «Пакета анализа» и «Поиска решения» можно привести решение задачи Марковица для нахождения портфеля ценных бумаг с минимальным риском. Эта задача решается в предметной области финансового менеджмента, в частности планирования инвестиций и оценки инвестиционных проектов. Но в данной работе используется только сервис «Поиск решения» в связи с тем, что решаемая задача имеет строго оптимизационный характер.

Подчеркивая актуальность использования Excel не только в качестве платформы для решения различного рода оптимизационных задач и автоматизаций вычислений, можно добавить, что он является незаменимым средством удобного представления данных с помощью таблиц, графиков, гистограмм и т.д. Таким образом, имеются сервисы,

позволяющие интегрировать Excel с базами данных (SaveToDB) и web-страницами в Microsoft SharePoint (Excel Services, Excel Mashup).

Apache POI – набор библиотек для Java, позволяющих работать с файлами формата Microsoft Office, таких как Word, PowerPoint и Excel. Библиотеки позволяют создавать, редактировать и считывать информацию с этих типов файлов, но не позволяют применять никакие дополнительные возможности данных приложений, в том числе и «Поиск решения». И метода в POI, работающего с данной надстройкой нет. Также есть возможность в самом Excel выполнить функции «Поиска решения» через макрос путем записи действий пользователя или через VBA-код. Но в POI даже нет метода, исполняющего какой-либо макрос, что очень странно и ограничивает возможности программиста для работы с Excel-файлами.

При создании приложений под Windows, часто используется следующий вариант обхода данной тупиковой ситуации: использование компонента Windows Script Host, который предназначен для запуска сценариев на скриптовых языках JScript и VBScript. Возможности сценариев на WSH значительно превосходят возможности командных .bat и .cmd-файлов: имеются полноценные языки с объектными возможностями, полный набор операций со строками, включая регулярные выражения, взаимодействие с любыми программами, реализующими объектный скриптовый интерфейс (Active Scripting или OLE Automation), доступ к методам и свойствам их объектов; операции с файлами и каталогами, обработка текстовых файлов, манипуляции с системным реестром и т. п.

Таким образом, используя возможности Windows Script Host, можно написать vbs-скрипт, который открывает нужный нам Excel-файл и исполняет в нем нужный макрос. Тем более, в данные сценарии можно передавать аргументы, как в bat-файлы.

Пример использования vbs-скрипта для вызова макроса:

```
Dim objXL
Set objXL = CreateObject('Excel.Application')
Set Args = WScript.Arguments
With objXL
    .Workbooks.Open (Args(0))
    .Application.Run Args(1)
    .Application.Quit
End With
Set objXL = Nothing
```

Здесь Args – список аргументов, передаваемых из вызываемой среды (Args(0) –путь до excel-файла, Args(1) –наименование макроса).

И данный vbs-скрипт можно вызвать из Java следующим образом:

```
Runtime.getRuntime.exec("cmd /c start " + vbs_scr + " " + path + " " + macros_name);
```

В последнем операторе vbs_scr – путь к vbs-скрипту, path – путь к excel-файлу, macros_name – наименование макроса, выполняющего функцию «Поиск решения». Все три переменные имеют тип String.

DOM (Document Object Model – объектная модель документа) – это независящий от платформы и языка программный интерфейс, позволяющий программам и скриптам получать доступ к содержимому HTML, XHTML и XML-документов, а также изменять содержимое, структуру и оформление таких документов.

Технология JDOM - это свободная Java-реализация DOM для XML, созданная с учётом особенностей языка и платформы Java. JDOM интегрируется с Document Object Model (DOM) и Simple API for XML (SAX), поддерживает XPath и XSLT. В JDOM используются внешние парсеры для генерации документов.

Но технология DOM содержит ряд ограничений, которые влекут за собой тяжелый (и в смысле использования памяти, и в смысле размера интерфейса) и громоздкий API, который бывает трудно изучать и использовать. Напротив, JDOM сформулирован как легкий API, прежде всего ориентированный на Java. JDOM отвечает перечисленным выше требованиям, следуя основным принципам DOM, но имеет свои особенности:

- JDOM специализирован для платформы Java. API использует, где возможно, встроенную в Java поддержку String, так что текстовые значения всегда доступны как String. Он также использует классы коллекций платформы Java 2, такие как List и Iterator, обеспечивая богатую среду для работы программистов, хорошо знакомых с языком Java.
- Нет иерархий. В JDOM элемент XML является экземпляром класса Element, атрибут XML является экземпляром класса Attribute, а сам XML-документ является экземпляром класса Document. Поскольку все они представляют разные концепции в XML, они всегда представляются собственными типами, а не как аморфные "узлы".
- Управляемый классом. Поскольку объекты JDOM являются непосредственными экземплярами таких классов, как Document, Element и Attribute,

создание их настолько же легко, насколько легко использование оператора new в языке Java.

Таким образом, с учетом вышеперечисленных особенностей, создание простейшего XML-документа с помощью JDOM выглядит следующим образом:

```
Element rootElement = new Element("metadata");  
rootElement.addContent(new org.jdom2.Comment("Indicate Excel cells for task"));  
Document doc = new Document(rootElement);  
Element ExcelPath = new Element("ExcelPath");  
ExcelPath.setAttribute("value", "");  
rootElement.addContent(ExcelPath);  
XMLOutputter outputter = new XMLOutputter();  
outputter.setFormat(Format.getPrettyFormat());  
FileOutputStream fw = new FileOutputStream(metadataXMLpath);  
outputter.output(doc, fw);  
fw.close();
```

В целом функционал разработанной программы осуществляется через следующие методы:

public static void CreateEmptyXML() – создает пустой xml-файл-метаданных, который должен заполнить пользователь.

public void CellsFromXML() – заполняет *Cell-поля класса адресами ячеек Excel, введенных пользователем, из xml-файла-метаданных.

public void CellFilling(Sheet sheet, Map<String, Double> values, Map<String, String> cells) – заполняет ячейки Excel по адресам из cells в листе sheet значениями из values.

public void CreateExcel(Runtime rt) – создает Excel файл, используя функции CellsFromXML и CellFilling. Rt – ссылка на объект среды исполнения, которая используется для вызова vbs-скрипта, который запускает макрос «Поиск решения».

public void AddToXML() throws IOException, JDOMException – записывает данные по решенной задаче в xml-файл-данных.

В процессе работы выяснилось, что JDOM технология является очень удобной для работы с xml-файлами в отличие от DOM, так как специально адаптирована под язык Java и использует отдельные типы для каждой составляющей XML-файла. К сожалению, было выяснено, что технология POI для работы с Excel-файлами обладает не слишком широким

функционалом, то есть не поддерживает выполнение макросов, а предназначена только для создания, считывания и редактирования Excel-файлов. Тем не менее, была реализована возможность выполнения макросов Excel из Java кода, используя vbs-скрипт.

Список использованных источников:

1. Филипов В.А., Щукин Б.А., Постояннов А.В., Язык XML и многомерная СУБД D3, издательство: Едиториал УРСС, 2001 г
2. <http://jakeroid.com/xml-v-java-s-primerami-jdom.html>
3. [3.http://www.jdom.org/downloads/index.html](http://www.jdom.org/downloads/index.html)
4. <http://poi.apache.org/>
5. [5.http://www.dissercat.com/content/predstavlenie-i-obrabotka-xml-baz-dannykh](http://www.dissercat.com/content/predstavlenie-i-obrabotka-xml-baz-dannykh)
6. <http://www.activexperts.com/activmonitor/windowsmanagement/scripts/msoffice/excel/>
7. <https://drive.google.com/file/d/0B6XQgaSuPWIGZi1NWIBHUHo3VjQ/edit?usp=sharing>